



QGIS

An Overview

Mitch Weier
North Dakota State Water Commission

QGIS

- Open-source GIS platform released under the General Public License (GPL)
 - Leading open source desktop GIS program
- Developed using Qt and C++
- Greatly improved over the past 4 years

QGIS

Strengths

- Free
- Universal (Windows, Linux, OS X, Android)
- Basic viewing, operations, and analysis
- Data format conversion
- Fast performance
- Advanced operations using plugins, python scripts, or using GRASS GIS as a backend
- Database interface
- Frequent updates, active development community

Weaknesses

- Cartographic tools and symbology lacking (high quality maps)
- Can be buggy, especially with new releases
- True 3D analysis and viewing

QGIS vs. Commercial GIS

QGIS vs. Commercial GIS

QGIS

- Free
- Performs simple tasks quickly
- Limited “off-the-shelf” options



gettyimages.com

Commercial GIS

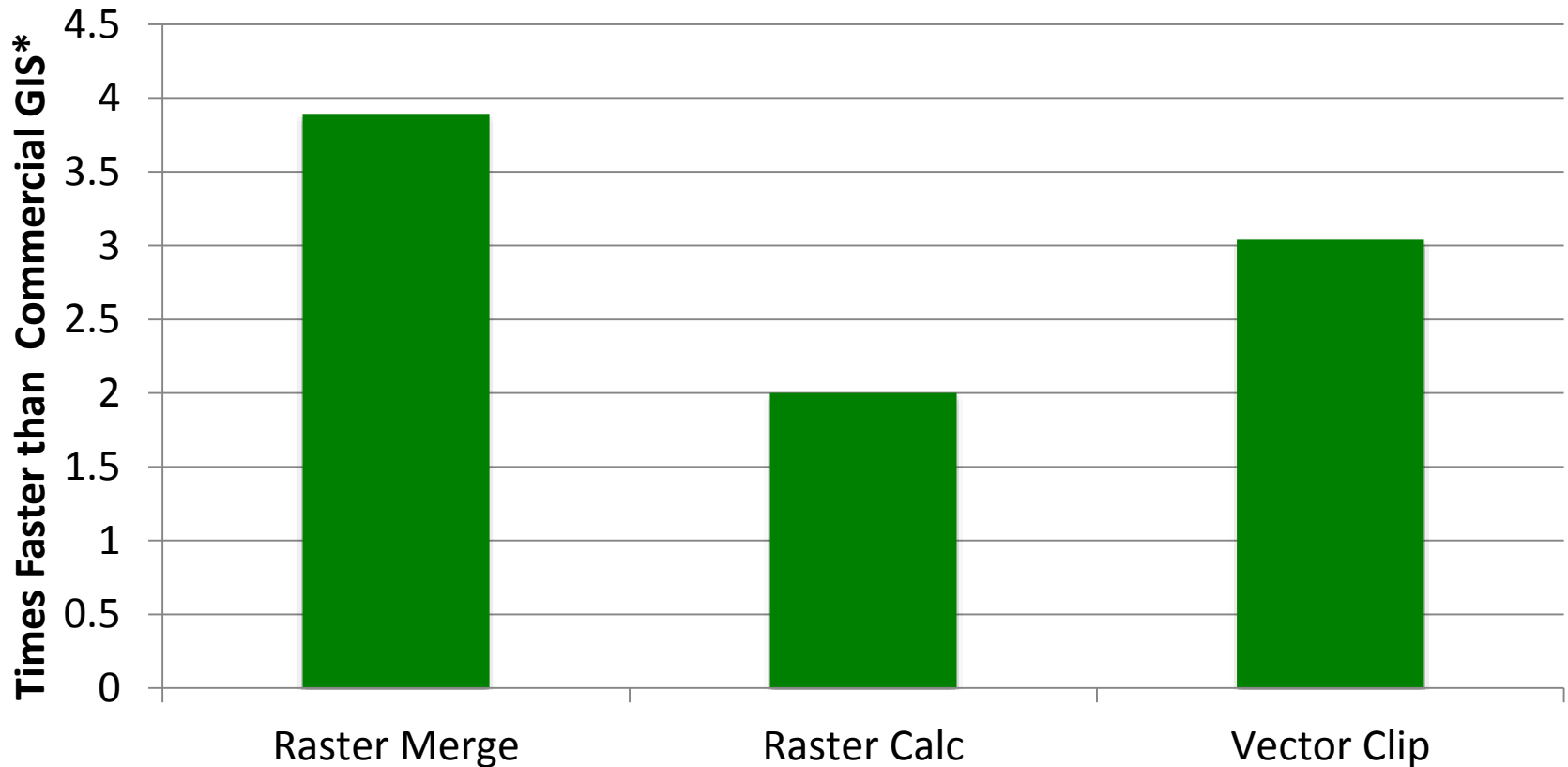
- Not free
- May or may not perform simple tasks quickly
- Many “off-the-shelf” options



quickmeme.com

Bench Test

QGIS 2.10 Performance



*Performance measured testing QGIS 2.10 and ArcMap 10.2.2 on a Windows 7 virtual machine using VMWare with one processor and 8 GB memory allocated

QGIS vs. Commercial GIS

QGIS

- Ability to perform complex tasks with plugins or with python scripts
- No tin support/3d viewing
- Adequate mapping capability for many users

Commercial GIS

- Ditto and then some
- Tin support and 3D viewing
- Advanced mapping capability

QGIS vs. Commercial GIS

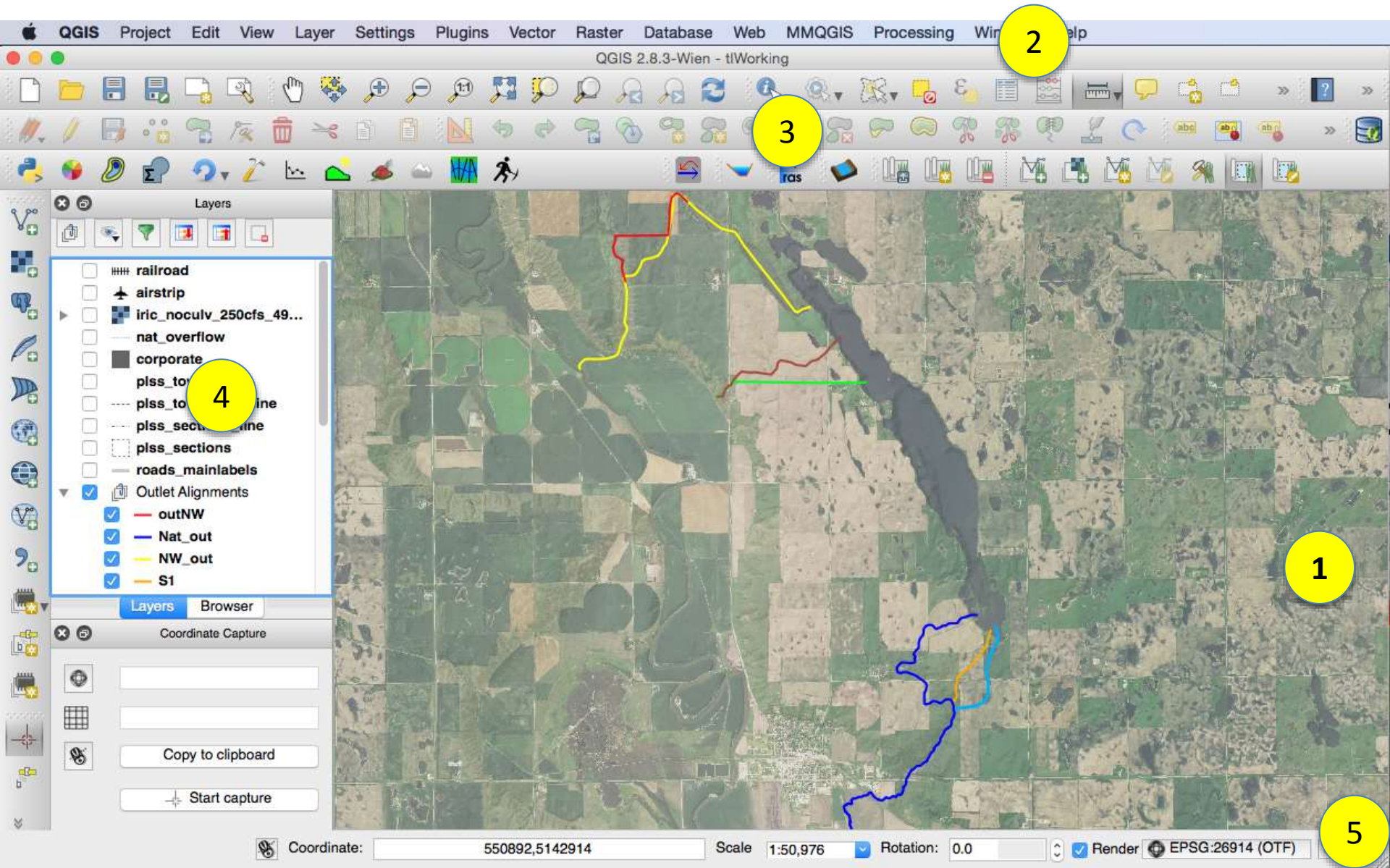
QGIS

- Crashes a lot
- Open source support system
 - Advanced users can troubleshoot errors by viewing code or function causing problems
 - Request help online from developers, bug report many times goes to original author
 - Much more responsive w/ payment!

Commercial GIS

- Crashes a lot
- Commercial support system

QGIS GUI Layout

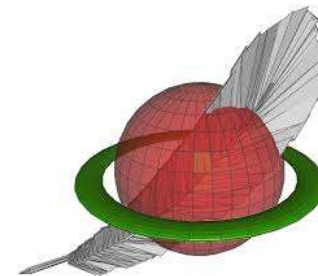


Internal Libraries and Backend Databases

- QGIS uses efficient internal libraries



- and useful backend databases

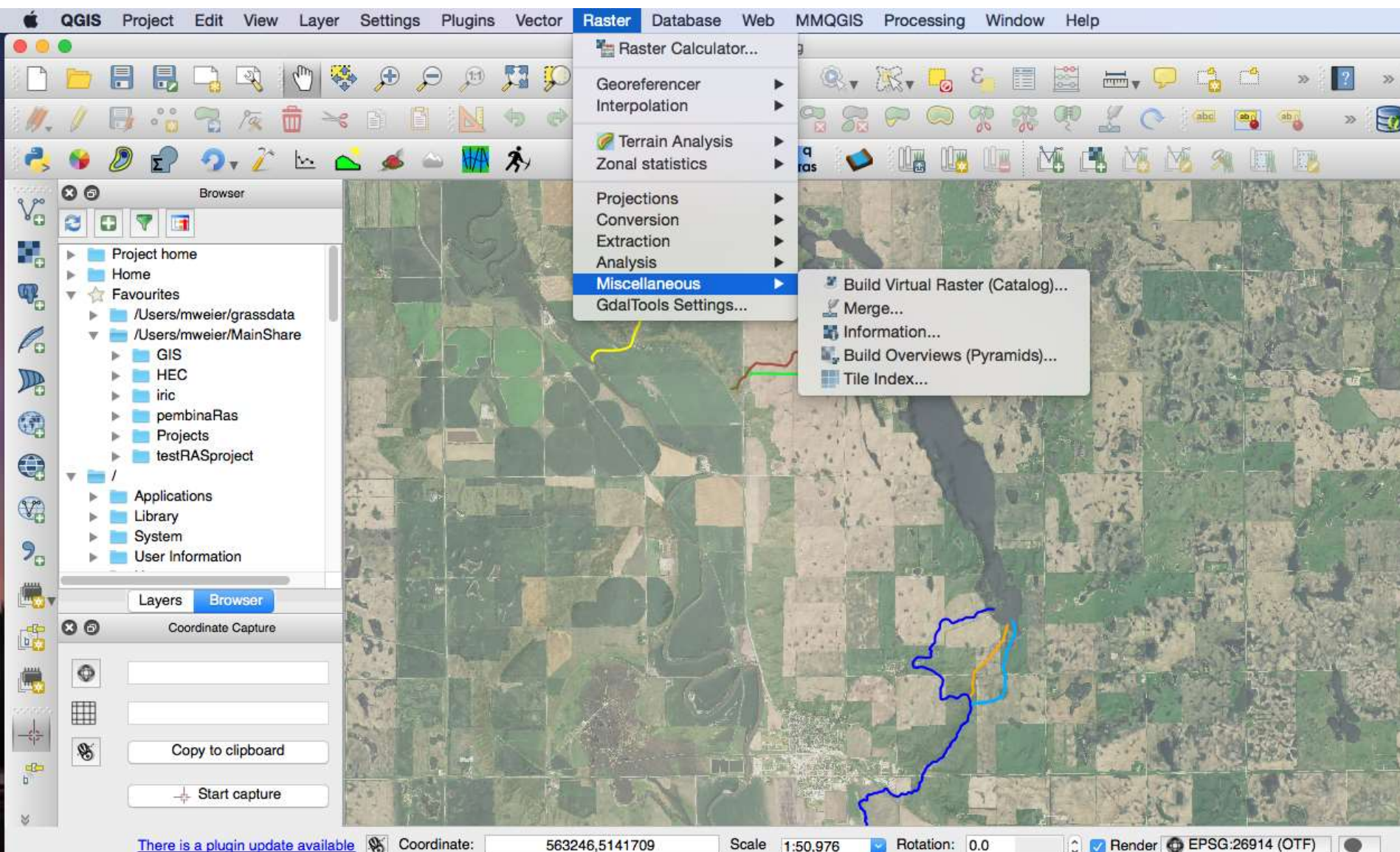


GDAL/Ogr

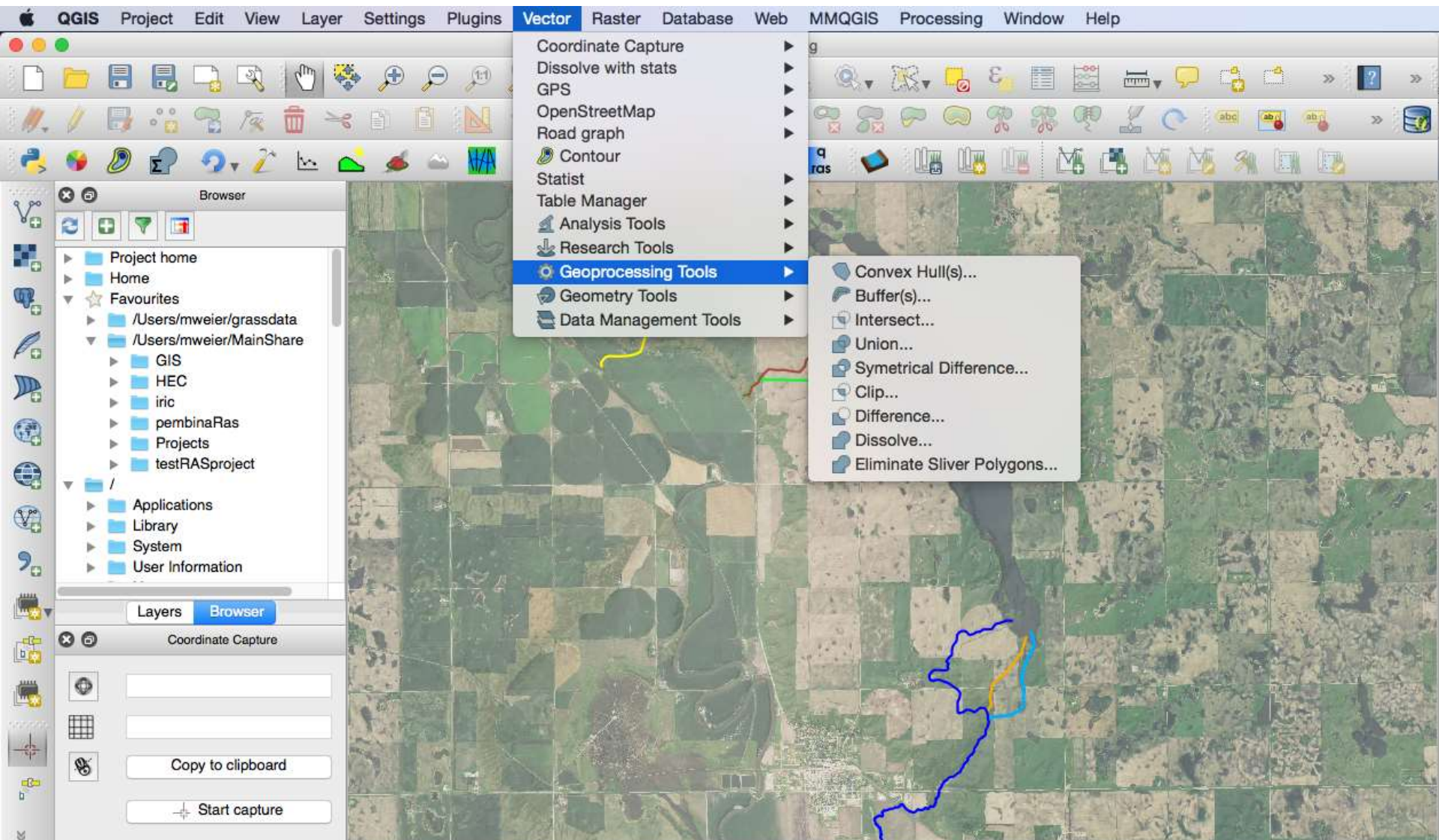


- QGIS relies on Geospatial Data Abstraction Library (GDAL)
 - GDAL is translator library for raster and vector geospatial data formats
 - Useful command line tools
 - GDAL Library supports over 140 raster formats
 - OGR Library supports over 80 vector formats
 - Also used in ArcGIS, GRASS GIS, Google Earth, SAGA GIS, R

GDAL (Raster) Tools

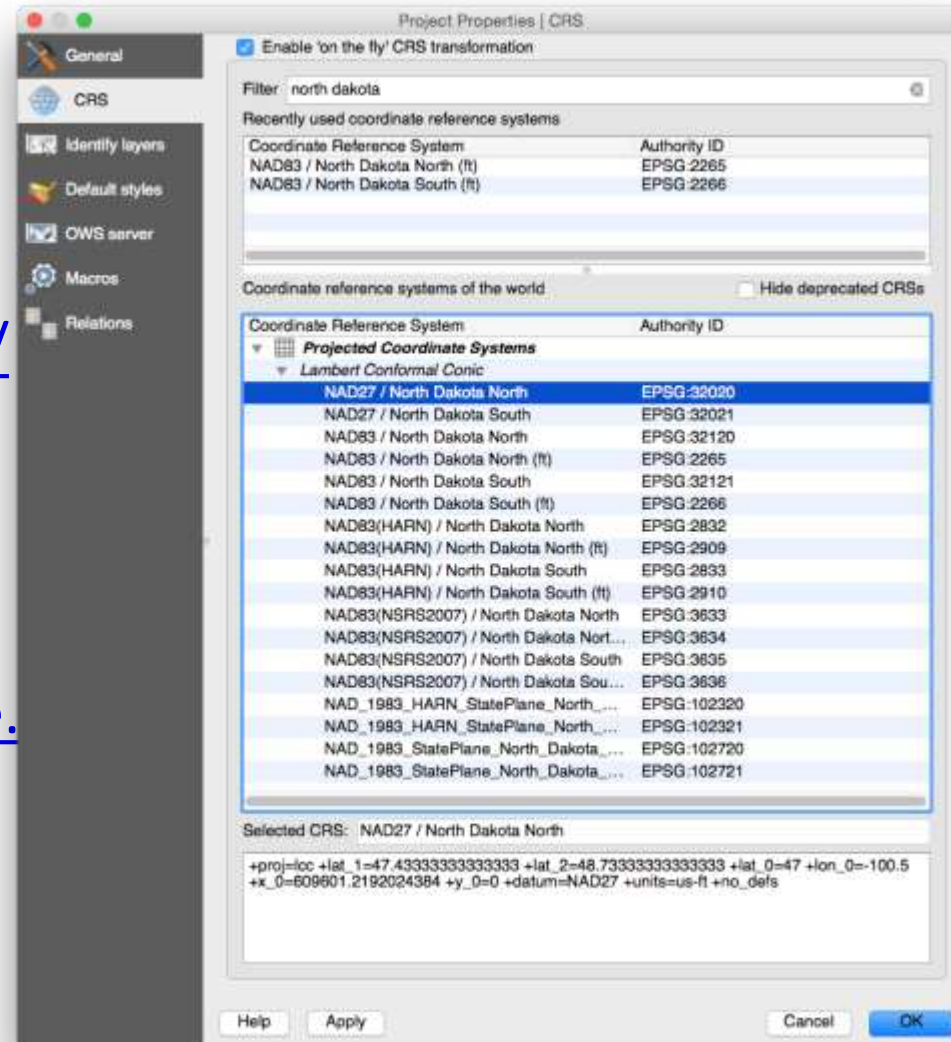


Vector Tools

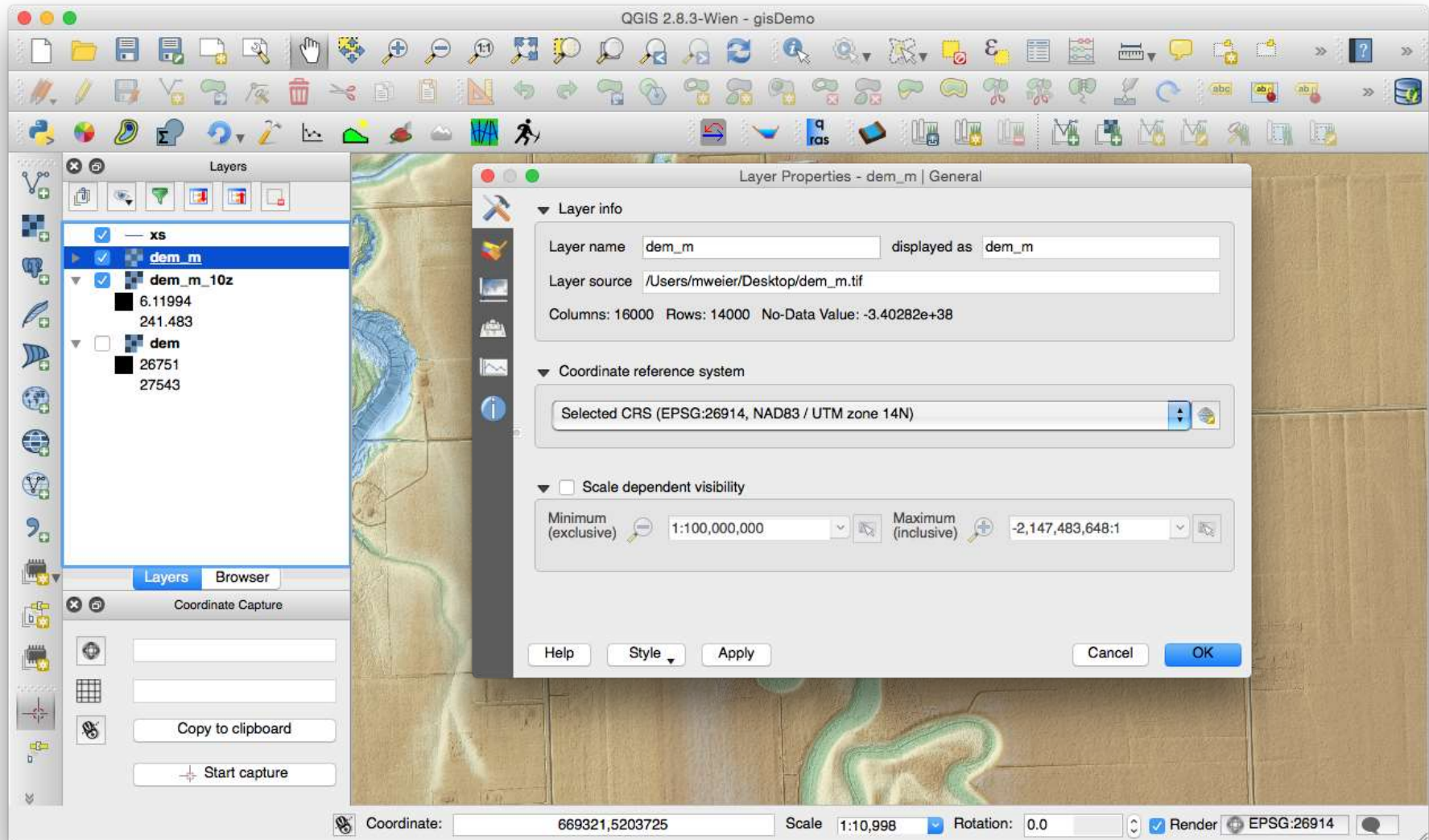


Projections/CRS

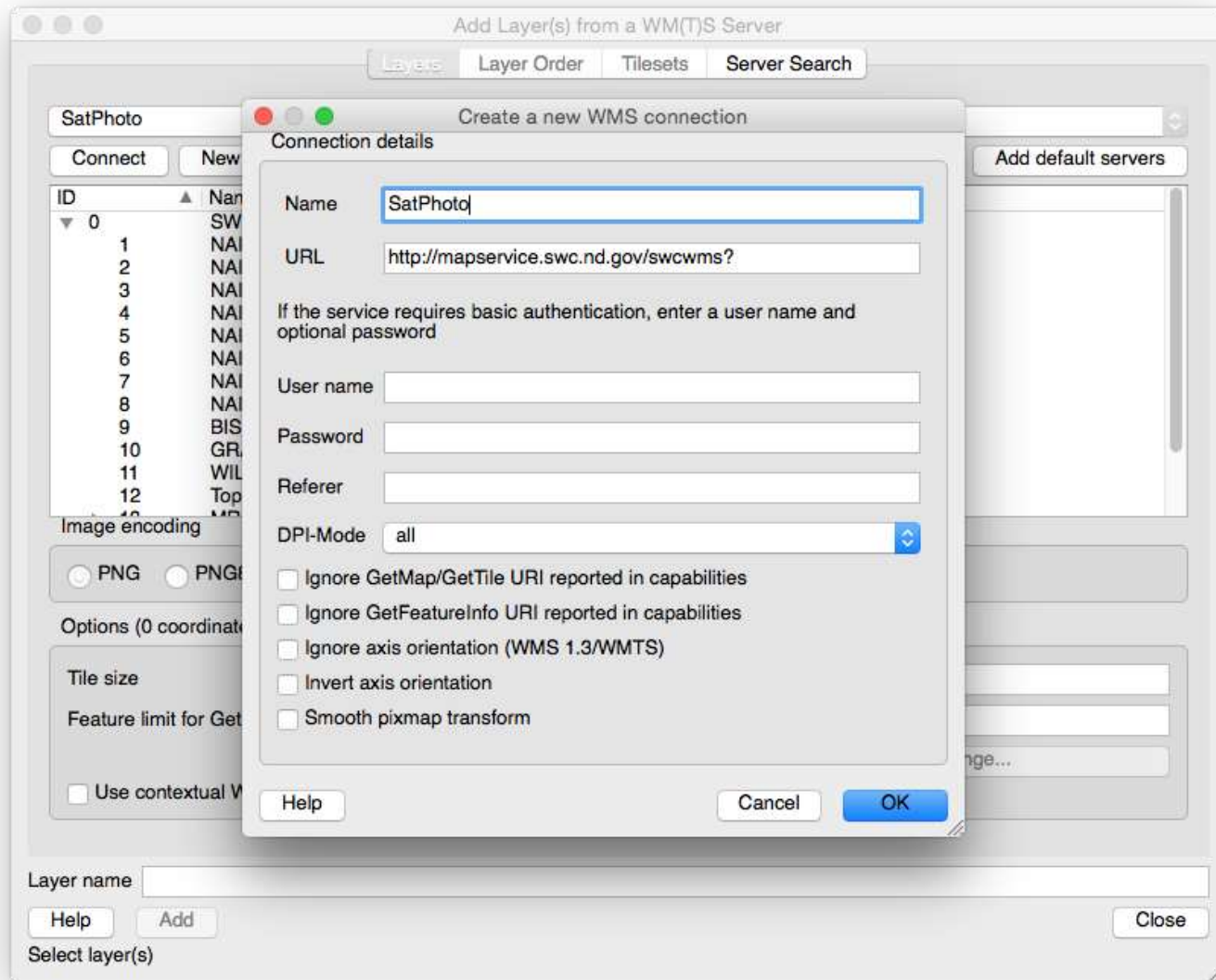
- Easy to keep track of
 - Proj4 framework
 - <https://trac.osgeo.org/proj/>
 - EPSG Geodetic Parameter Set Codes
 - <http://spatialreference.org>



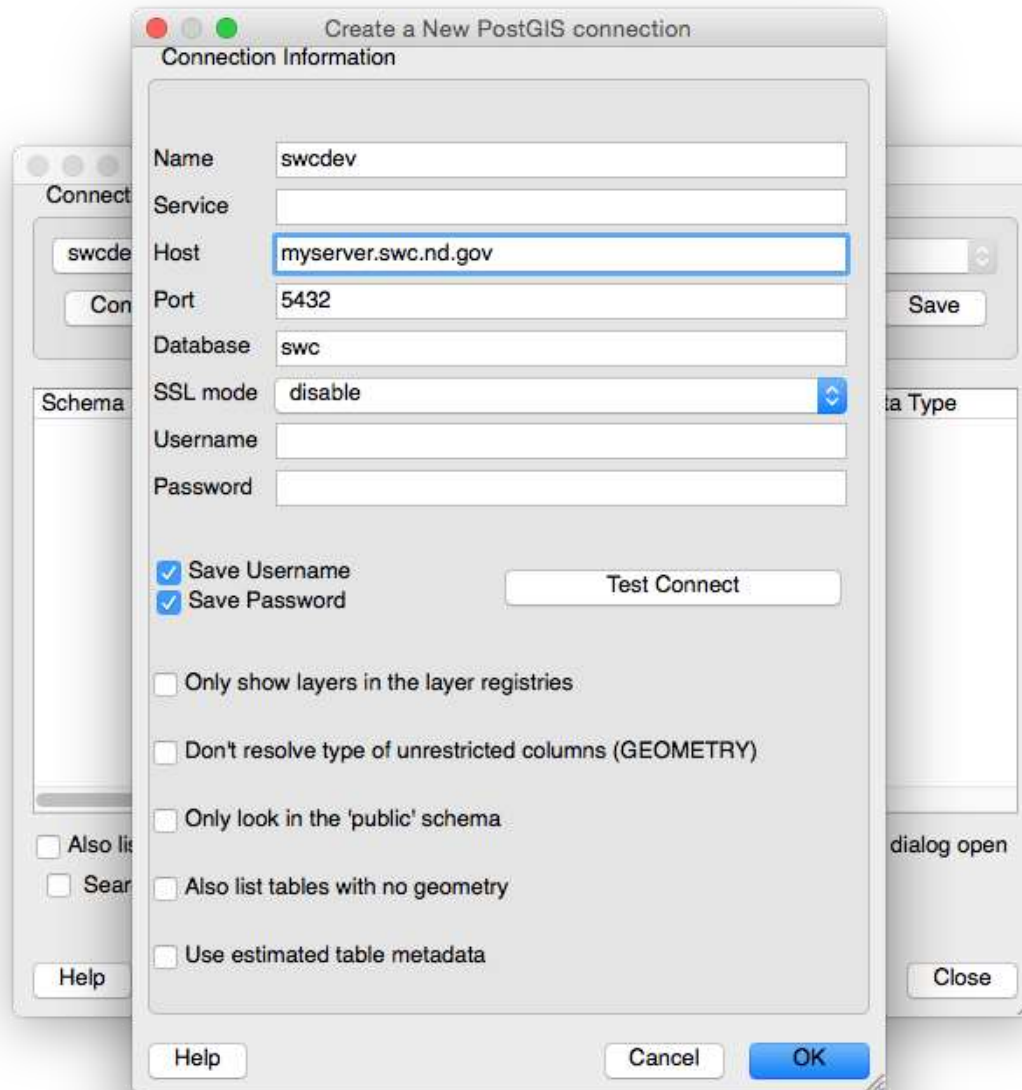
Projections



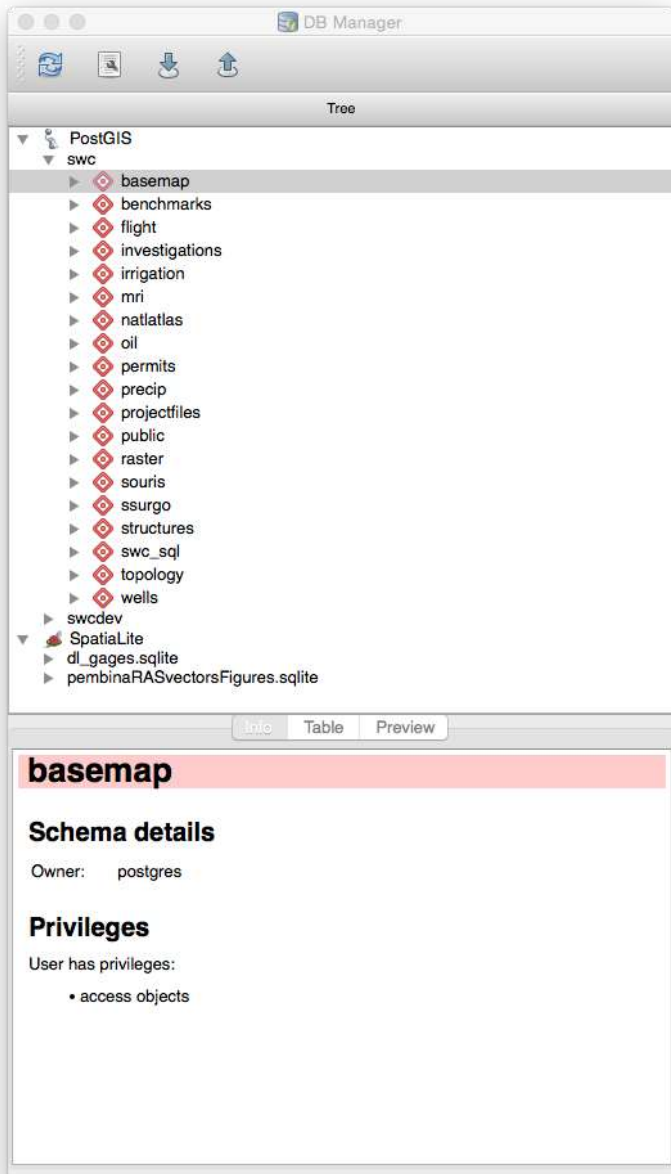
WMS Connections



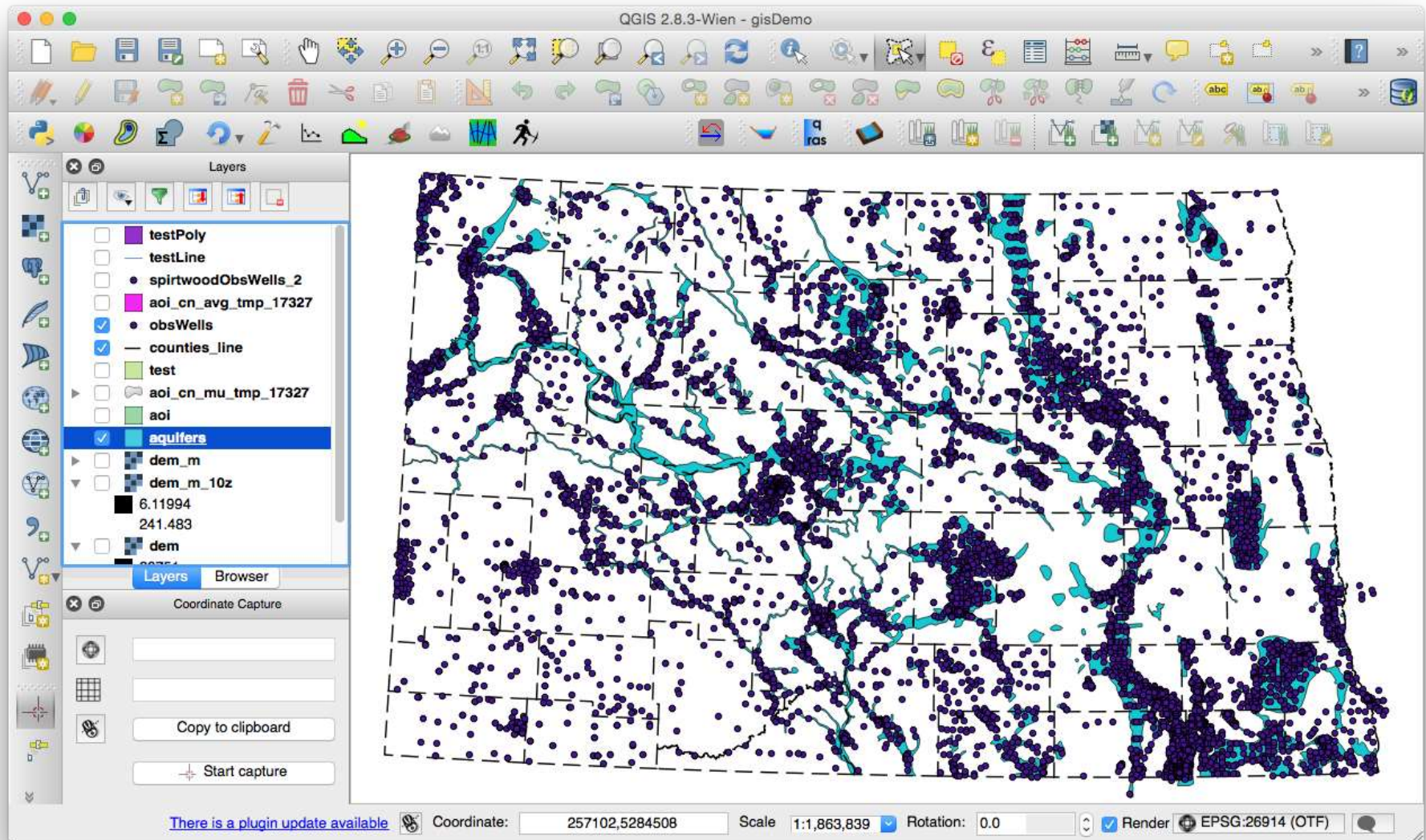
Database Connections



DB Manager



DB Manager



DB Manager

SQL window - swc [PostGIS]

SQL query: wells wells Store Delete

```
SELECT well_index, site_location, name, t1.the_geom as the_geom
FROM wells.well_header t1, basemap.aquifers t2
WHERE name ILIKE '%Spiritwood%' AND purpose ILIKE 'Observation%' AND ST_INTERSECTS(t1.the_geom, t2.the_geom)
```

Execute (F5) 1175 rows, 0.0 seconds Clear

Result:

	well_index	site_location	name	the_geom
1	967	12905307BB...	Spiritwood	0101000020...
2	969	12905309AA...	Spiritwood	0101000020...
3	978	12905507CCC	Spiritwood	0101000020...

☒ Load as new layer

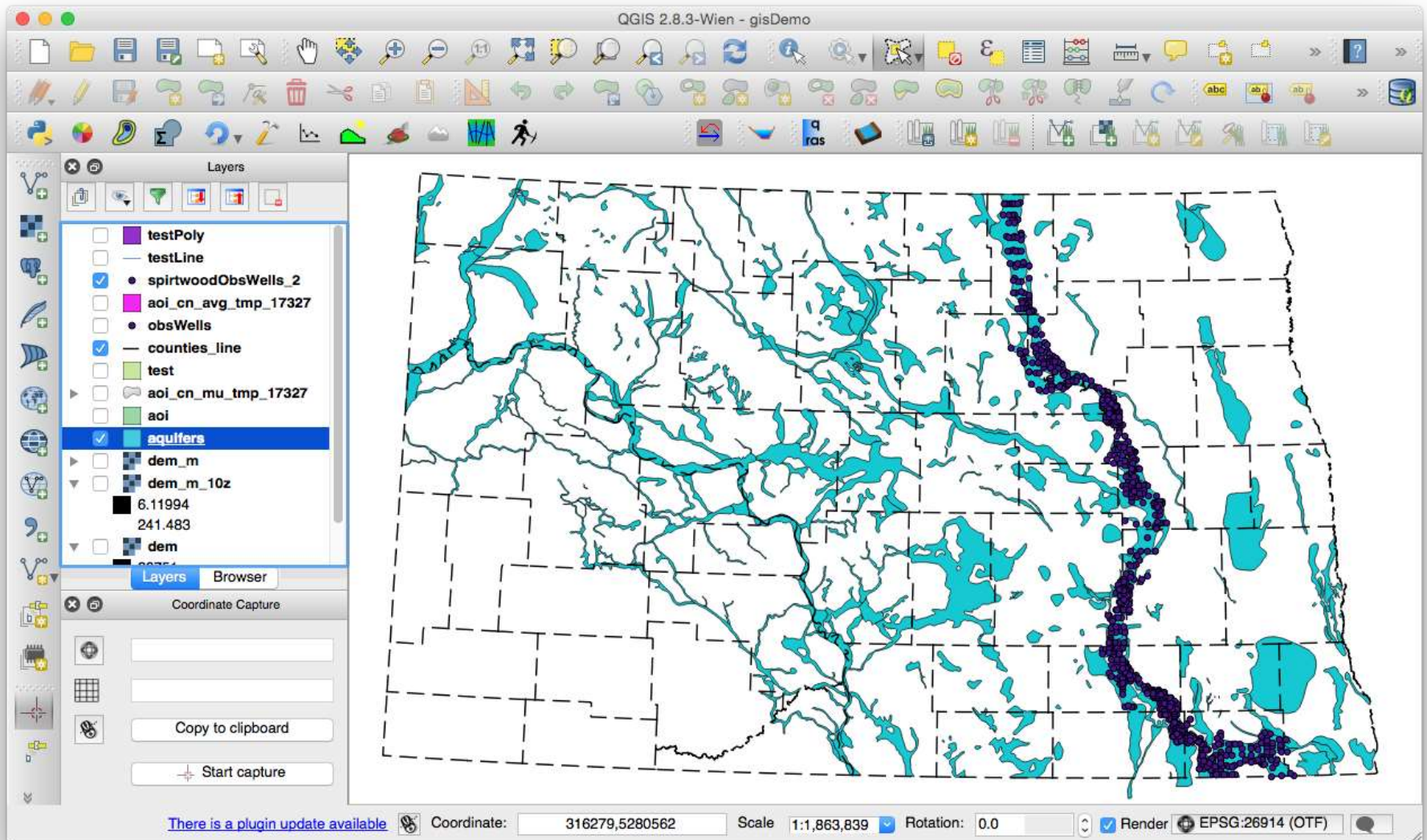
Column with unique integer values well_index Geometry column the_geom Retrieve columns

Layer name (prefix) spirtwoodObsWells Load now!

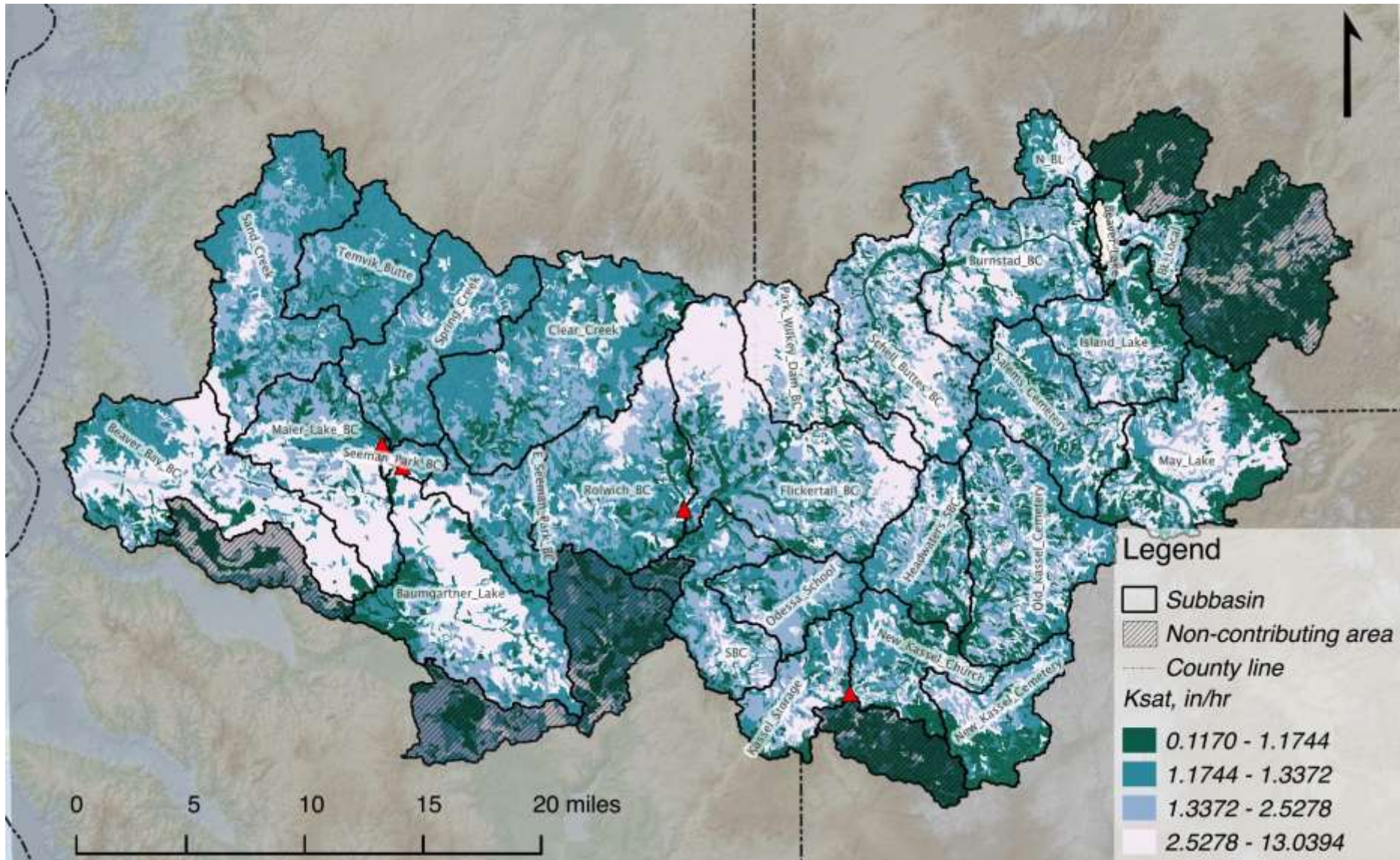
☐ Avoid selecting by feature id

Close

DB Manager



DB Manager



Plugins

The screenshot shows the QGIS Plugins Manager window. The left sidebar has a filter menu with 'All' selected, and other options: 'Installed', 'Not installed', 'Upgradeable', and 'Settings'. The main list of plugins is displayed, with 'Profile tool' selected and highlighted in blue. The right pane shows the details for the 'Profile tool' plugin.

Plugins | All (461)

Search

All

Installed

Not installed

Upgradeable

Settings

Processing LWGEOM Provider

Processing PS Tools provider

Processing Workflows

ProcessingPermaclim provider

☒ Profile from Points

☒ **Profile tool**

Proportional circles

PS Time Series Viewer

pyArchInit

pyUPVBib

☒ **Q-RAS**

QBAN(O)

QChainage

QConsolidate

QGIS Cloud Plugin

QGIS Remote Control

Qgis Web Connector

qgis2leaf

Qgis2threejs

qgis2web

qgis_epanet

QGISCartoDB

QGISConefor

qgisio

QgisMarkerCluster Plugin

qgSurf

QgsWcpsClient1

Help

Profile tool

Plots terrain profile

This tool plots profile lines from raster layers. Supports multiple lines as well as export to svg, pdf, png or csv file. Works with both Qwt5 and Matplotlib plotting libraries.

★★★★☆ 66 rating vote(s), 79071 downloads

Category: Raster
Tags: raster, profile
More info: [homepage](#) [tracker](#) [code repository](#)

Author: [Borys Jurgiel](#) - [Patrice Verchere](#) - [Etienne Tourigny](#)

Installed version: 3.6.7 (in /Users/mweier/.qgis2/python/plugins/profiletool)
Available version: 3.6.7 (in QGIS Official Plugin Repository)

changelog:

3.6.7 : Add option to include coordinates to the csv output (by martst)
3.6.6 : Adjustable Y scale, proper representation of nodata values in Qwt5 plots
3.6.5 : Refresh the plot when layer's dataChanged signal is emitted (by Martin Dobias)
3.6.4 : bugfixes from radosuav
3.6.3 : fix bugs #6870, #9002 and #9111
3.6.2 : fix bugs #8890, #8945 and again #6679
3.6.1 : support for plugin layers for QGIS API 2 (by Peter Wells)
3.6.0 : update to sip api v2 (and qgis 2.0)

3.5.6 : fix bugs #8890, #8945 and again #6679

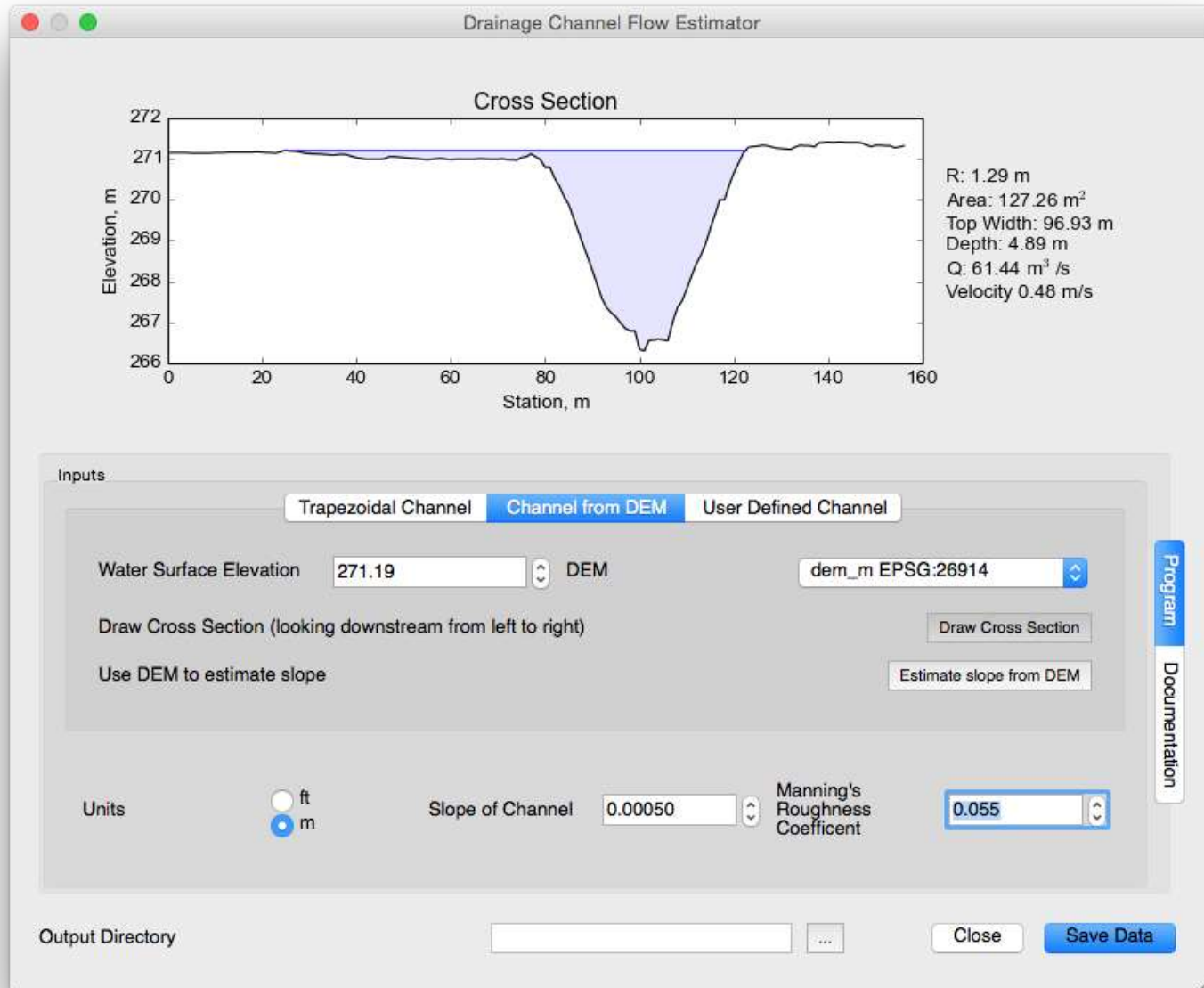
Upgrade all

Uninstall plugin

Reinstall plugin

Close

Custom Plugins



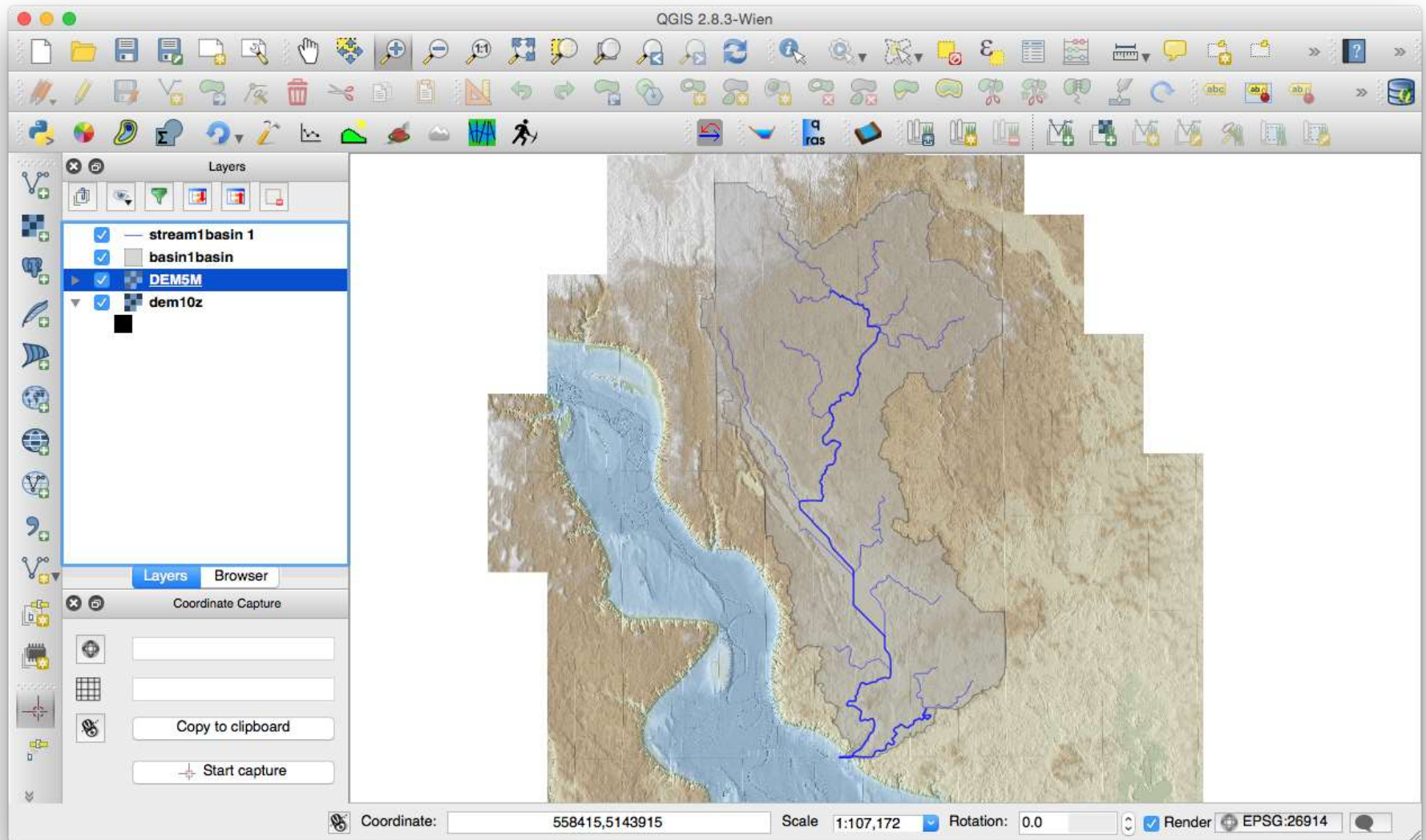
Python Console

The screenshot displays the QGIS 2.8.3-Wien - gisDemo interface. The main map area shows a topographic map with a yellow rectangle highlighting a specific region. The left sidebar contains the 'Layers' panel, which lists several layers: testLine, spirtwoodObsWells_2, aoi_cn_avg_tmp_17327, obsWells, counties_line, test, aoi_cn_mu_tmp_17327, aoi, aquifers, dem_m, dem_m_10z, 6.11994, 241.483, dem, and 26751. The right sidebar contains the 'Python Console' window, which shows the following code and output:

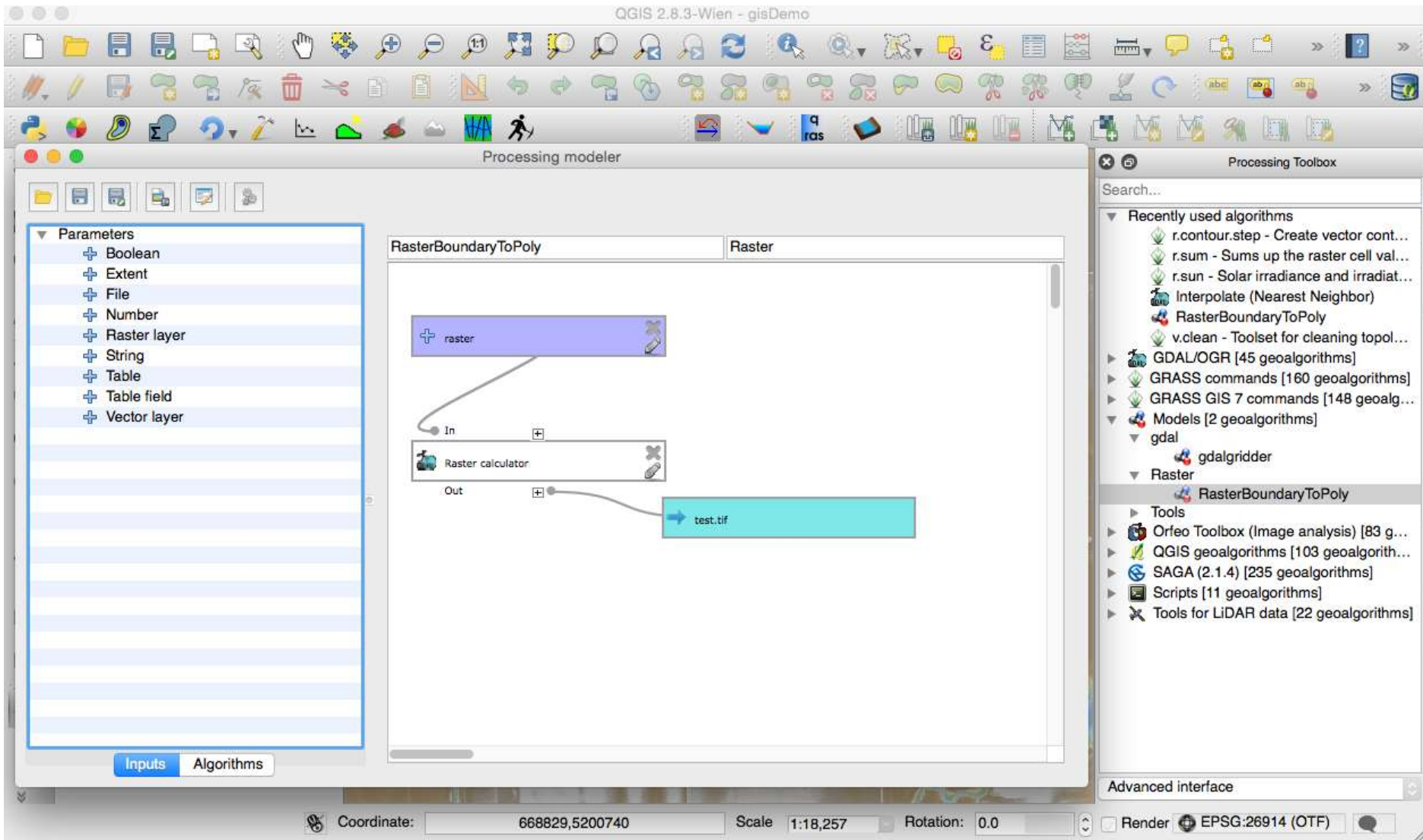
```
1 Python Console
2 Use iface to access QGIS API interface or Type help(
3   iface) for more info
4 >>> lineToXY()
5 665785.610039 5203608.50092
6 666070.055455 5204168.77219
7 >>> polyToXY()
8 663531.663889 5203266.48531
9 665135.44358 5203327.38834
10 665206.49711 5201703.30764
11 663562.115402 5201632.25411
12 663531.663889 5203266.48531
```

The bottom status bar indicates '1 feature(s) selected on [There is a plugin update available](#)'. The coordinate is 668079,5199622, the scale is 1:28,773, and the rotation is 0.0. The render button is checked, and the EPSG:26914 (OTF) coordinate system is selected.

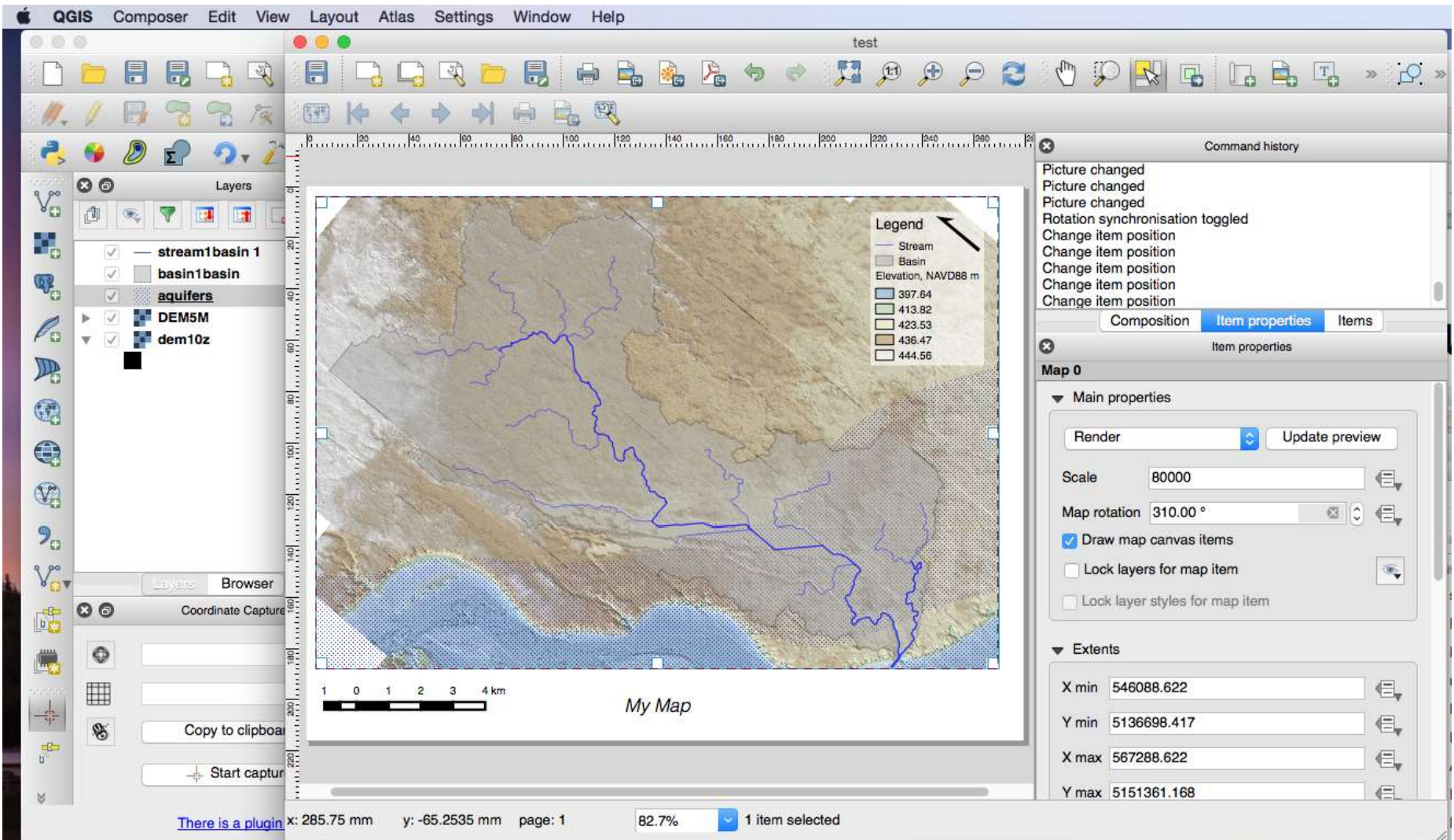
GRASS GIS Plugin



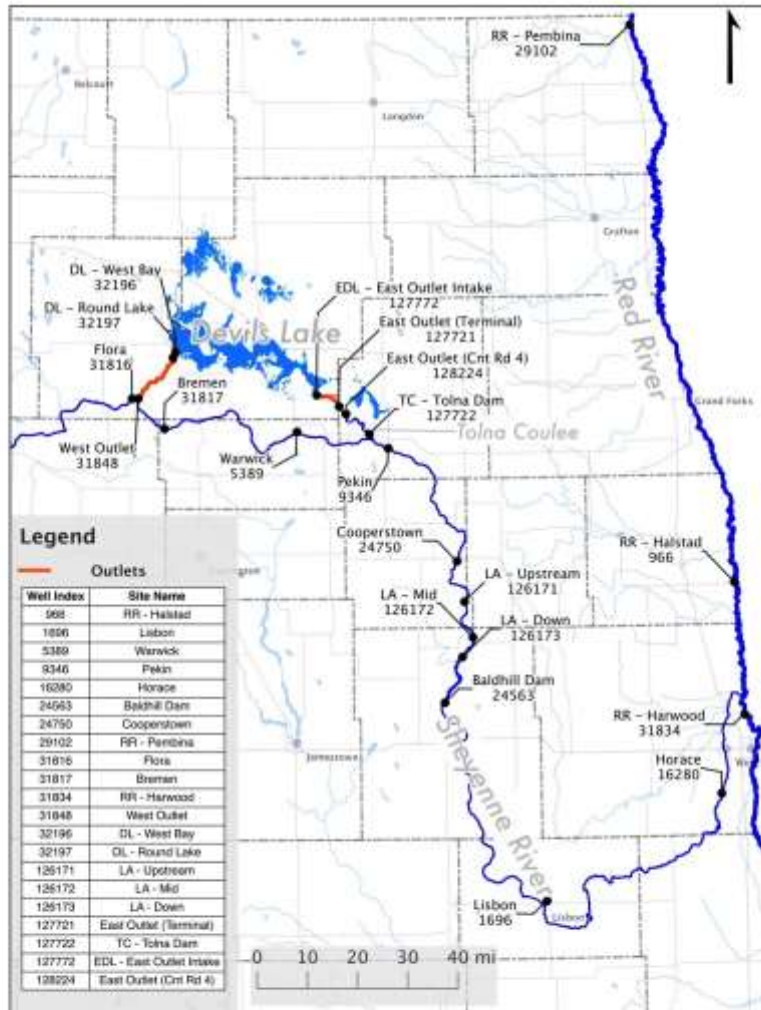
Processing Toolbox



Print Composer

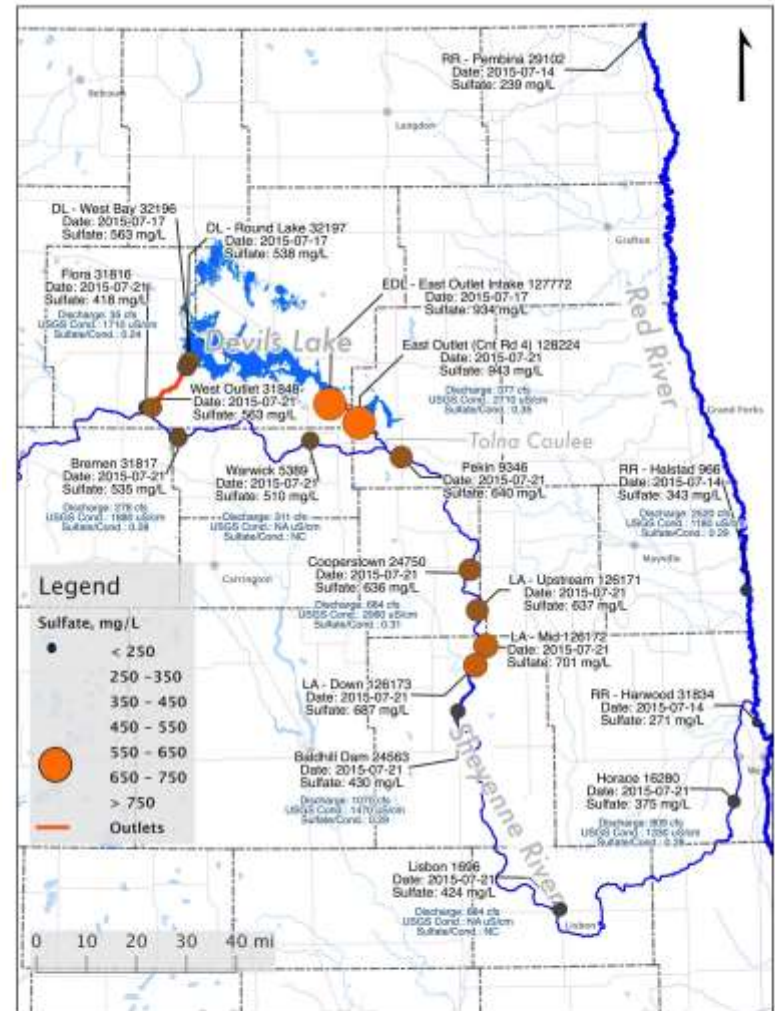


Print Composer



Devils Lake Outlet Water Quality Monitoring Locations

2015-03-11



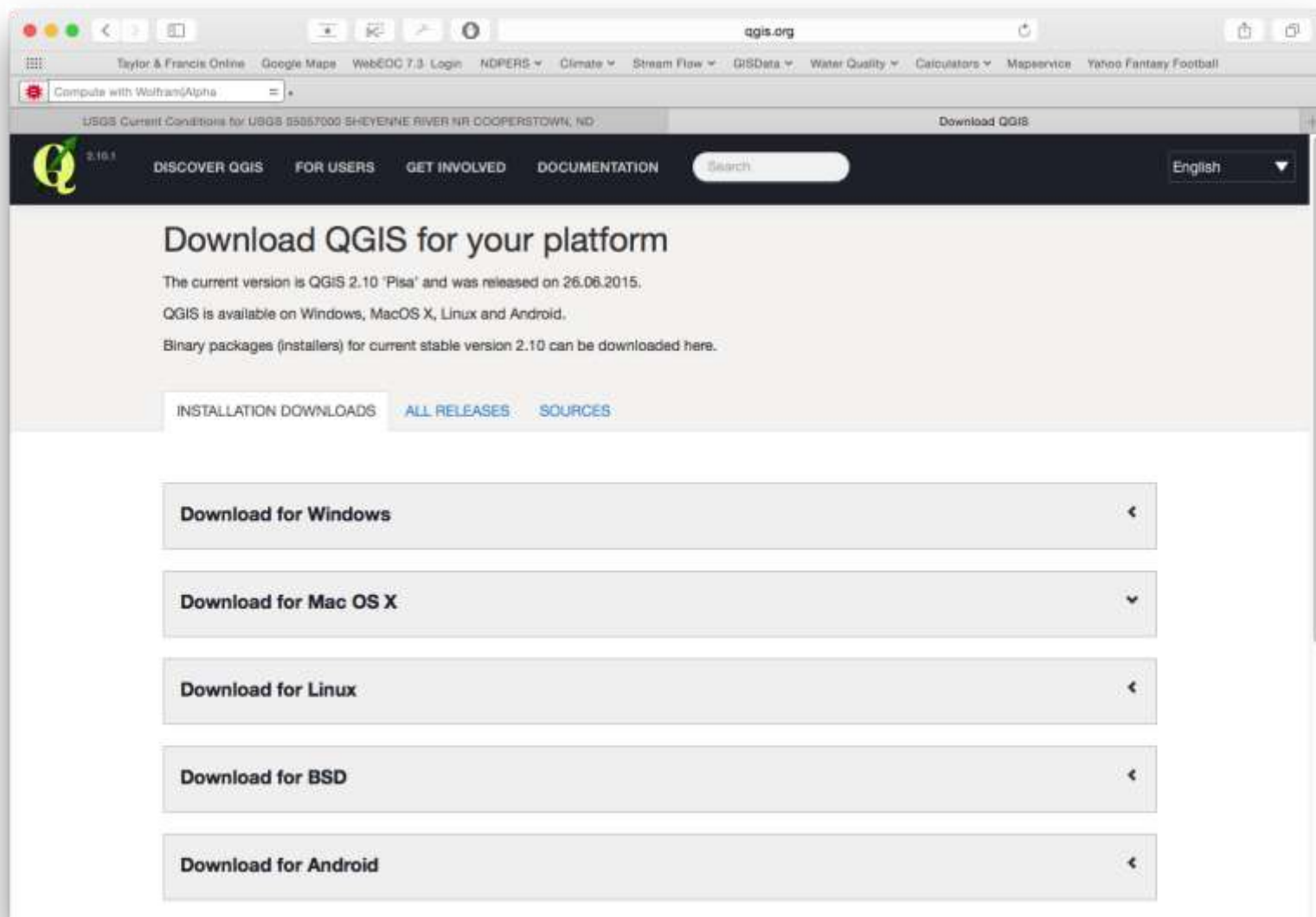
Devils Lake Outlet Operations Preliminary Sulfate Concentrations 07/29/2015

State Water Commission

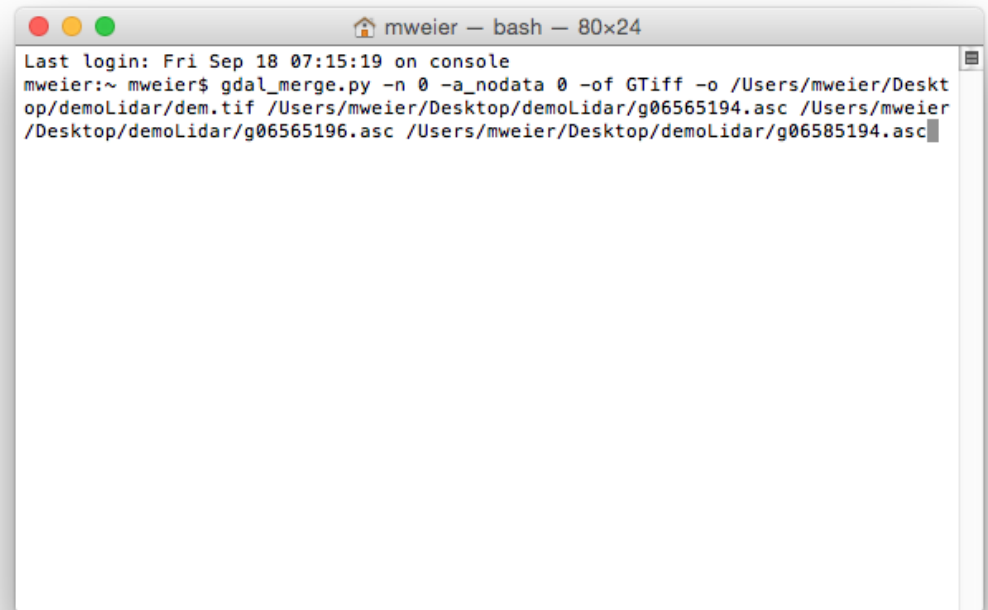
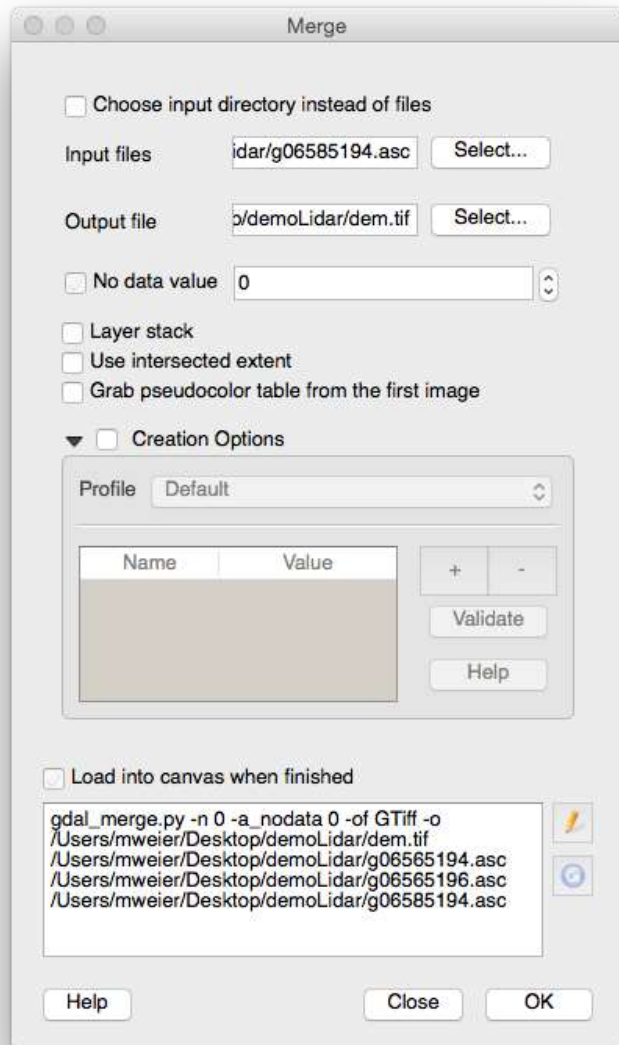
Summary

- QGIS is a simpler program than commercial GIS programs, but what it does, it does well
- Maybe most useful for beginners and advanced users

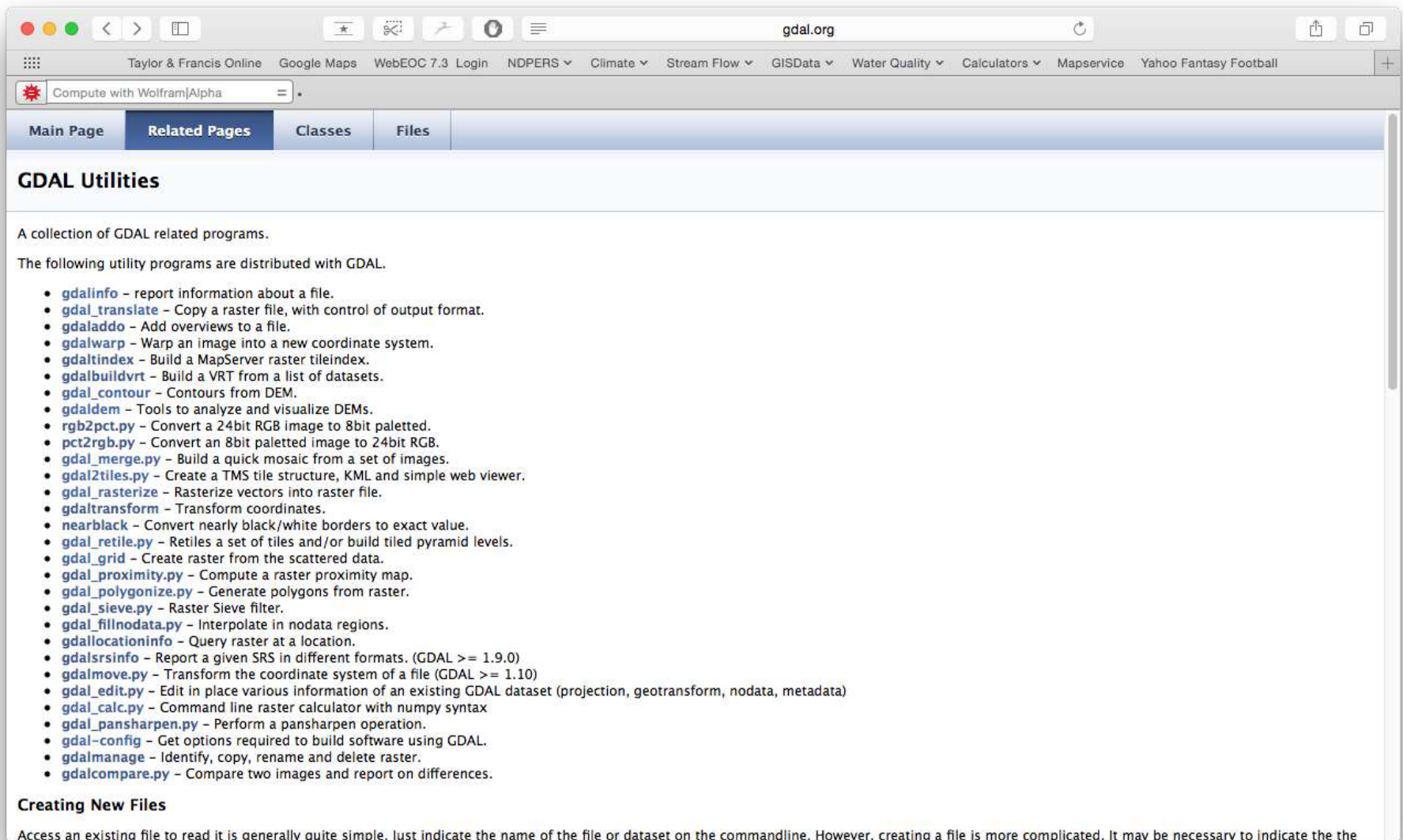
Available at qgis.org



GDAL (Raster) Tools



GDAL (Raster) Tools



The screenshot shows a web browser window with the URL `gdal.org`. The browser's address bar and tabs are visible at the top. The website's navigation menu includes links to 'Main Page', 'Related Pages', 'Classes', and 'Files'. The 'Related Pages' tab is selected, displaying the 'GDAL Utilities' section. This section describes a collection of GDAL-related programs and lists 30 specific utility programs with their functions. Below the list, there is a section titled 'Creating New Files' which provides information on how to access existing files and create new ones using the command line.

GDAL Utilities

A collection of GDAL related programs.

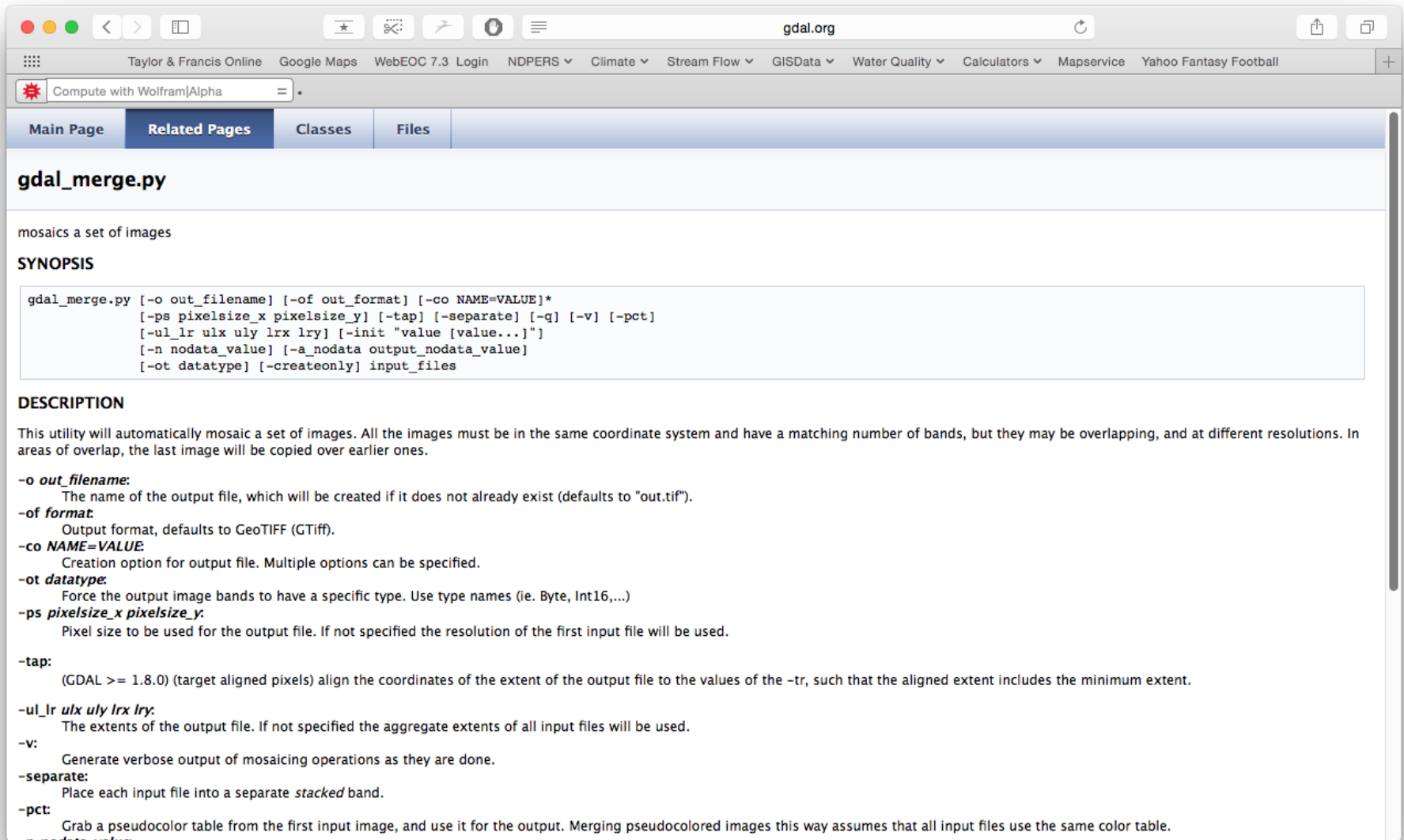
The following utility programs are distributed with GDAL.

- [gdalinfo](#) – report information about a file.
- [gdal_translate](#) – Copy a raster file, with control of output format.
- [gdaladdo](#) – Add overviews to a file.
- [gdalwarp](#) – Warp an image into a new coordinate system.
- [gdaltindex](#) – Build a MapServer raster tileindex.
- [gdalbuildvrt](#) – Build a VRT from a list of datasets.
- [gdal_contour](#) – Contours from DEM.
- [gdaldem](#) – Tools to analyze and visualize DEMs.
- [rgb2pct.py](#) – Convert a 24bit RGB image to 8bit paletted.
- [pct2rgb.py](#) – Convert an 8bit paletted image to 24bit RGB.
- [gdal_merge.py](#) – Build a quick mosaic from a set of images.
- [gdal2tiles.py](#) – Create a TMS tile structure, KML and simple web viewer.
- [gdal_rasterize](#) – Rasterize vectors into raster file.
- [gdaltransform](#) – Transform coordinates.
- [nearblack](#) – Convert nearly black/white borders to exact value.
- [gdal_retile.py](#) – Retiles a set of tiles and/or build tiled pyramid levels.
- [gdal_grid](#) – Create raster from the scattered data.
- [gdal_proximity.py](#) – Compute a raster proximity map.
- [gdal_polygonize.py](#) – Generate polygons from raster.
- [gdal_sieve.py](#) – Raster Sieve filter.
- [gdal_fillnodata.py](#) – Interpolate in nodata regions.
- [gdallocationinfo](#) – Query raster at a location.
- [gdalsrsinfo](#) – Report a given SRS in different formats. (GDAL >= 1.9.0)
- [gdalmove.py](#) – Transform the coordinate system of a file (GDAL >= 1.10)
- [gdal_edit.py](#) – Edit in place various information of an existing GDAL dataset (projection, geotransform, nodata, metadata)
- [gdal_calc.py](#) – Command line raster calculator with numpy syntax
- [gdal_pansharpen.py](#) – Perform a pansharpen operation.
- [gdal-config](#) – Get options required to build software using GDAL.
- [gdalmanage](#) – Identify, copy, rename and delete raster.
- [gdalcompare.py](#) – Compare two images and report on differences.

Creating New Files

Access an existing file to read it is generally quite simple. Just indicate the name of the file or dataset on the commandline. However, creating a file is more complicated. It may be necessary to indicate the the

GDAL (Raster) Tools



The screenshot shows a web browser window with the URL `gdal.org`. The browser's address bar and tabs are visible at the top. Below the browser window, the website's navigation menu includes links for `Main Page`, `Related Pages` (which is highlighted), `Classes`, and `Files`. The main content area is titled `gdal_merge.py` and contains the following information:

mosaics a set of images

SYNOPSIS

```
gdal_merge.py [-o out_filename] [-of out_format] [-co NAME=VALUE]*
               [-ps pixelsize_x pixelsize_y] [-tap] [-separate] [-q] [-v] [-pct]
               [-ul_lr ulx uly lrx lry] [-init "value [value...]" ]
               [-n nodata_value] [-a_nodata output_nodata_value]
               [-ot datatype] [-createonly] input_files
```

DESCRIPTION

This utility will automatically mosaic a set of images. All the images must be in the same coordinate system and have a matching number of bands, but they may be overlapping, and at different resolutions. In areas of overlap, the last image will be copied over earlier ones.

-o out_filename:
The name of the output file, which will be created if it does not already exist (defaults to "out.tif").

-of format:
Output format, defaults to GeoTIFF (GTiff).

-co NAME=VALUE:
Creation option for output file. Multiple options can be specified.

-ot datatype:
Force the output image bands to have a specific type. Use type names (ie. Byte, Int16,...)

-ps pixelsize_x pixelsize_y:
Pixel size to be used for the output file. If not specified the resolution of the first input file will be used.

-tap:
(GDAL >= 1.8.0) (target aligned pixels) align the coordinates of the extent of the output file to the values of the -tr, such that the aligned extent includes the minimum extent.

-ul_lr ulx uly lrx lry:
The extents of the output file. If not specified the aggregate extents of all input files will be used.

-v:
Generate verbose output of mosaicing operations as they are done.

-separate:
Place each input file into a separate *stacked* band.

-pct:
Grab a pseudocolor table from the first input image, and use it for the output. Merging pseudocolored images this way assumes that all input files use the same color table.